In the Claims

The following listing of the claims replaces all previous listings.

Please cancel claims 2, 8, and 14.

Please amend claims 1, 3, 7, 9, and 13 as follows.

1. (Currently Amended) A method for filtering one or more messages for transmission to a subscriber computing system according to an individual information request criteria, the method comprising:

constructing a binary decision diagram implication graph for each individual information request criteria specified for each subscriber, the binary decision diagram implication graph including a plurality of nodes expressing each information request criteria in an if-then-else normal form, the nodes being evaluated until a logical true or false node is reached;

identifying logical implications from one or more nodes in a binary decision diagram from a first information request criteria to one or more corresponding binary decision diagrams within a second information request criteria;

receiving one or more messages to be filtered;

evaluating a first information request criteria based upon information within the received messages;

evaluating one or more information request criteria based upon information within the received messages using the identified logical implications between one or more binary decision diagrams within the information request criteria being evaluated and one or more binary decision diagrams previously evaluated; and

transmitting the received message to the subscriber computing system corresponding to an information request criteria evaluated to be satisfied by information contained within the received message.

2. (Canceled)

3. (Currently Amended) The method according to claim [[2]] 1, wherein the constructing step further comprises:

recursively visiting the high and low successors for each node in the binary decision diagrams;

while visiting each node, determine the precondition pre(X') for each successor and compute the target t(X') for all visited nodes and apply permissible implications; and

iterate the processing for all implications.

4. (Original) The method according to claim 3, wherein permissible implications for a node M with successor node N include:

if node N is equal to the high successor high(M), and

if p(M) implies p(N), then remove N and set the high(M) equal to high(N); and

if p(M) implies ! p(N), then remove N and set high(M) equal to low(N).

5. (Original) The method according to claim 3, wherein permissible implications for a node M with successor node N include:

if node N is equal to the low successor low(M), and

if ! p(M) implies p(N), then remove N and set the low(M) equal to high(N); and

if ! p(M) implies ! p(N), then remove N and set low(M) equal to low(N).

6. (Original) The method according to claim 5, wherein the evaluating steps further comprises:

determining if a current node is a leaf node in the binary decision diagram;

if the current node is a leaf node, marking the information request criteria as being decided and returning the value of the current node;

if the current node is not a leaf node, determining a value of the expression for the node p(X);

if the value of the expression of the node is true

setting X' = high(X) otherwise X' = low(X);

inserting X' into the rank; and

visiting the targets of node X' to compare the current node with the target node;

if the target node is lower than the current node according to a predicate order, update the current node.

7. (Currently Amended) A computer program product readable by a computing system and encoding instructions for filtering one or more messages to be transmitted to a subscriber computing system according to an individual information request criteria, the computing process comprising:

constructing a binary decision diagram implication graph for each individual information request criteria specified for each subscriber, the binary decision diagram implication graph including a plurality of nodes expressing each information request criteria in an if-then-else normal form, the nodes being evaluated until a logical true or false node is reached;

identifying logical implications from one or more nodes in a binary decision diagram from a first information request criteria to one or more corresponding binary decision diagrams within a second information request criteria;

receiving one or more messages to be filtered;

evaluating a first information request criteria based upon information within the received messages;

evaluating one or more information request criteria based upon information within the received messages using the identified logical implications between one or more binary decision diagrams within the information request criteria being evaluated and one or more binary decision diagrams previously evaluated; and

transmitting the received message to the subscriber computing system corresponding to an information request criteria evaluated to be satisfied by information contained within the received message.

8. (Canceled)

9. (Currently Amended) The computer program product according to claim [[8]] 7, wherein the constructing step further comprises:

recursively visiting the high and low successors for each node in the binary decision diagrams;

while visiting each node, determine the precondition pre(X') for each successor and compute the target t(X') for all visited nodes and apply permissible implications; and

iterate the processing for all implications.

10. (Original) The computer program product according to claim 9, wherein permissible implications for a node M with successor node N include:

if node N is equal to the high successor high(M), and

if p(M) implies p(N), then remove N and set the high(M) equal to high(N); and

if p(M) implies ! p(N), then remove N and set high(M) equal to low(N).

11. (Original) The computer program product according to claim 9, wherein permissible implications for a node M with successor node N include:

if node N is equal to the low successor low(M), and

if ! p(M) implies p(N), then remove N and set the low(M) equal to high(N); and

if ! p(M) implies ! p(N), then remove N and set low(M) equal to low(N).

12. (Original) The computer program product according to claim 9, wherein the evaluating steps further comprises:

determining if a current node is a leaf node in the binary decision diagram;

if the current node is a leaf node, marking the information request criteria as being decided and returning the value of the current node;

if the current node is not a leaf node, determining a value of the expression for the node p(X);

if the value of the expression of the node is true

setting X' = high(X) otherwise X' = low(X);

inserting X' into the rank; and

visiting the targets of node X' to compare the current node with the target node;

if the target node is lower than the current node according to a predicate order, update the current node.

13. (Currently Amended) A publication-subscription broker server computing system for filtering one or more messages to be transmitted to a subscriber computing system according to an individual information request criteria, the broker server computing system comprises:

a memory module;

a mass storage system; and

a programmable processing module, the programmable processing module performing a sequence of operations to implement the following:

constructing a binary decision diagram implication graph for each individual information request criteria specified for each subscriber, the binary decision diagram implication graph including a plurality of nodes expressing each information request criteria in an if-then-else normal form, the nodes being evaluated until a logical true or false node is reached;

identifying logical implications from one or more nodes in a binary decision diagram from a first information request criteria to one or more corresponding binary decision diagrams within a second information request criteria;

receiving one or more messages to be filtered;

evaluating a first information request criteria based upon information within the received messages;

evaluating one or more information request criteria based upon information within the received messages using the identified logical implications between one or more binary decision diagrams within the information request criteria being evaluated and one or more binary decision diagrams previously evaluated; and

transmitting the received message to the subscriber computing system corresponding to an information request criteria evaluated to be satisfied by information contained within the received message.

14. (Canceled)

15. (Original) The broker server computing system according to claim 13, wherein the constructing the implication graph further comprises:

recursively visiting the high and low successors for each node in the binary decision diagrams;

while visiting each node, determine the precondition pre(X') for each successor and compute the target t(X') for all visited nodes and apply permissible implications; and

iterate the processing for all implications.

16. (Original) The broker server computing system according to claim 14, wherein permissible implications for a node M with successor node N include:

if node N is equal to the high successor high(M), and

if p(M) implies p(N), then remove N and set the high(M) equal to high(N); and

if p(M) implies ! p(N), then remove N and set high(M) equal to low(N).

17. (Original) The broker server computing system according to claim 15, wherein permissible implications for a node M with successor node N include:

if node N is equal to the low successor low(M), and

if ! p(M) implies p(N), then remove N and set the low(M) equal to high(N); and

if ! p(M) implies ! p(N), then remove N and set low(M) equal to low(N).

18. (Original) The broker server computing system according to claim 15, wherein the evaluating steps further comprises:

determining if a current node is a leaf node in the binary decision diagram;

if the current node is a leaf node, marking the information request criteria as being decided and returning the value of the current node;

if the current node is not a leaf node, determining a value of the expression for the node p(X);

if the value of the expression of the node is true

setting X' = high(X) otherwise X' = low(X);

inserting X' into the rank; and

visiting the targets of node X' to compare the current node with the target node;

if the target node is lower than the current node according to a predicate order, update the current node.